AI-Powered Document Interaction: A Natural Language Processing Approach For Conversational PDF Analysis

Dr.P.Sumathi[1], R.Swathiramya[2], Karthik A[3], Mukilesh Kumar S[4], Nivash S[5], Nithyasri R[6]
[1, 2, 3, 4, 5, 6] Dept of Artificial Intelligence and Data Science
[1, 2, 3, 4, 5, 6] SNS College of Engineering Coimbatore, Tamil Nadu, India

Abstract- *The Chat with PDF is an innovative solution designed to simplify the process of querying multiple PDF documents through a conversational AI interface. With the increasing volume of digital documents in PDF format, especially in academic, corporate, and legal domains, there is a growing need for a system that enables users to efficiently access information without manually searching through large files. This project addresses that need by using state-of-the-art natural language processing (NLP) models to allow users to ask questions and receive contextually relevant answers from multiple PDFs simultaneously.*

The core of this system is built on transformer-based models, such as sentence-transformers and Hugging Face's transformers, which enable the embedding of PDF content into vector space, making it possible to perform efficient semantic searches. The user uploads PDFs via a web interface built with Streamlit, where the documents are processed, and their content is embedded using pre-trained models from the sentence-transformers library. These embeddings are then queried with natural language inputs, and the system retrieves and ranks the most relevant sections from the PDFs, providing a concise and accurate answer to the user's question.

This allows users to access previous conversations and maintain a long-term record of queries and responses. The web application also leverages PyPDF2 for extracting text from PDF files, making it adaptable to various document types and structures.

Keywords- Multi-PDF Chatbot,Natural Language Processing, PDF Embedding, Semantic Search, Streamlit Interface, Document Querying, Text Extraction, Sentence Transformers, Conversational AI, Information Retrieval, Contextual Understanding, User-Friendly Chatbot, Knowledge Management.

I. INTRODUCTION

In today's digital age, vast amounts of information are stored in PDF format, particularly in fields like academia, law, business, and research. These documents often contain crucial information, but manually searching through multiple PDFs for specific details can be time-consuming and inefficient. Traditional keyword search methods, while helpful, are often limited as they may fail to capture the context or provide accurate, concise responses to complex queries.

The Chat with pdf is a novel solution designed to address this challenge by leveraging state-of-the-art natural language processing (NLP) technologies to enable conversational querying of multiple PDF documents. The system allows users to ask questions in natural language, and the AI-powered chatbot retrieves relevant sections from the uploaded PDFs, offering contextually accurate answers. This conversational interface significantly enhances user experience by providing a more intuitive and efficient way to interact with large document datasets.

Built using Streamlit for the front end, the project integrates cutting-edge models from the sentence-transformers and transformers libraries to embed and process PDF content. These models convert the document text into vector representations, allowing semantic searches rather than just keyword-based retrieval.

This project is particularly useful for researchers, students, and professionals who need to extract information from multiple documents quickly and efficiently. It eliminates the need for manual searches and provides a seamless way to obtain specific information through a conversational interface. The combination of advanced AI models and a user-friendly interface makes the Multi-PDF Chat App a powerful tool for interacting with and extracting knowledge from PDF documents.

II. EXISTING SYSTEM

The current landscape of document search and retrieval is largely dominated by keyword-based search tools found in traditional PDF readers or search engines. These tools allow users to search for specific words or phrases within a single document, but they are often limited by their inability to understand the context or meaning behind the query. As a result, they may return results that are either too broad or too narrow, forcing users to manually sift through irrelevant content to find the information they need. Furthermore, these systems are generally limited to searching one document at a time, making them inefficient for users working with large collections of PDFs, such as academic researchers, legal professionals, or business executives. Even advanced enterprise search engines, which may offer some degree of cross-document search, still rely heavily on exact keyword matches and often fail to capture the nuances of user queries, especially in complex, multi-document scenarios. These limitations create a significant gap in usability, particularly when users are seeking specific, contextually relevant information from multiple sources. Additionally, traditional systems lack a conversational interface that allows users to interact with the documents naturally. There is also minimal support for organizing and retrieving past searches or conversations, leading to repetitive and time-consuming workflows. In short, existing systems are not equipped to handle the dynamic, context-rich queries that many users require when dealing with extensive document repositories.

2.1 Key Issues

1. Limited Search Capabilities: Traditional PDF readers offer only basic keyword-based searches, which lack contextual understanding. This often results in irrelevant results, especially when dealing with multiple documents.
2. Manual Document Navigation: Users have to open, read, and search through each document individually, which is time-consuming and inefficient for users handling multiple PDFs.
3. Inability to Process Multiple PDFs Simultaneously: Existing systems typically do not allow users to query across multiple PDF files at once, making it difficult to gather information from multiple sources in a unified manner.
4. Lack of Conversational Interaction: Most PDF viewers lack a conversational interface, making it challenging for users to interact with the content dynamically or ask questions in natural language.

2.2 Objective

The primary objective of the Chat with PDF is to develop an AI-powered system that allows users to interact with multiple PDF documents through a conversational interface. This system is designed to retrieve specific, contextually relevant information from the documents in response to user queries. The key objectives include:

1. Efficient Document Querying: To enable users to query multiple PDFs simultaneously and get precise, relevant responses, reducing the time spent on manual searches.
2. Natural Language Interaction: To provide an intuitive, conversational interface where users can ask questions in natural language, and the AI system will return accurate answers from the PDFs.
3. Contextual Understanding: To leverage transformer models for semantic understanding, ensuring that the system captures context beyond simple keyword matches.
4. Scalability: To store document embeddings and chat histories in a scalable manner, allowing users to retrieve previous conversations and ensuring persistence of document knowledge.

2.3 Scope

1. Integration with Transformer Models: Utilizing transformer-based models, such as sentence-transformers and transformers from Hugging Face, for embedding PDF content and enabling natural language processing for accurate query results.
2. Web Application Development: Building an interactive web interface using Streamlit for users to upload PDFs, ask queries, and receive responses.
3. Document Embedding and Retrieval: Implementing document embedding mechanisms to convert PDF content into vector space, allowing the system to perform semantic searches across multiple documents.
4. PDF Text Extraction: Leveraging libraries like PyPDF2 to extract content from PDFs, allowing the system to handle various document structures and formats.

III. SOFTWARE REQUIRMENTS

3.1 Programming Language

Python serves as the primary programming language for the "Chat with Multiple PDFs" project due to its versatility, simplicity, and extensive library ecosystem. Python’s readability and ease of use make it an ideal choice for building complex applications involving natural language processing (NLP) and data handling. With its vast array of libraries, Python facilitates tasks such as text extraction from PDFs, embedding generation, document querying, and conversational interactions. Leveraging libraries like PyPDF2 for PDF handling, sentence-transformers for semantic embeddings, and transformers for NLP model integration, Python enables the creation of a powerful, user-friendly chatbot capable of retrieving relevant information across multiple PDFs. This functionality empowers users to efficiently extract insights from large sets of documents, enhancing productivity and supporting informed decision-making across various domains.

3.2 System Requirements

RAM: 4GB or 8GB
Windows 10
Intel Core i5/i7 processor
At least 60 GB of Usable Hard Disk Space

3.3 Libraries

1.STREAMLIT:

Streamlit is an open-source Python framework that simplifies the process of building and deploying interactive web applications for data science and machine learning projects. With Streamlit, developers can quickly create visually appealing apps using just a few lines of Python code, without requiring extensive web development knowledge. Streamlit's focus on ease of use, automatic updates upon code changes, and seamless integration with machine learning models makes it a powerful tool for creating dashboards, data exploration tools, and interactive interfaces for machine learning workflows, all while requiring minimal effort to set up and deploy.

2.LANGCHAIN:

LangChain is a powerful framework designed for building applications that integrate large language models (LLMs) with external data sources and APIs. It simplifies the process of chaining together multiple AI tasks, enabling developers to create sophisticated, multi-step pipelines for tasks like question-answering, document retrieval, and conversational agents. LangChain excels in managing the interactions between LLMs and tools like databases, search engines, and APIs, making it ideal for applications that require real-time data or complex workflows. Its modular architecture allows for flexibility and scalability, making it a popular choice for AI-driven app development.

3. FAISS:

FAISS (Facebook AI Similarity Search) is an open-source library developed by Facebook AI, designed for efficient similarity search and clustering of high-dimensional vectors. It is optimized for large-scale datasets and is highly useful in applications like nearest neighbor search, document retrieval, and recommendation systems. FAISS supports both exact and approximate nearest neighbor searches and is optimized for performance on CPUs, with additional support for GPU acceleration. The CPU version of FAISS is particularly suited for environments without GPU resources, providing efficient indexing and querying capabilities by leveraging multi-threading and vectorization techniques. This makes it a powerful tool for AI-driven tasks like embedding search and semantic similarity calculations in NLP and other fields.

4. PyPDF2:

PyPDF2 is a popular Python library used for manipulating and working with PDF files. It allows users to perform a variety of tasks such as reading, merging, splitting, and rotating PDFs, as well as extracting text and metadata from PDF documents. PyPDF2 is capable of handling both encrypted and password-protected PDFs, making it a versatile tool for managing PDF content programmatically. Though it lacks some advanced features like handling complex layouts or converting PDFs to other formats (e.g., images or text files), PyPDF2 remains a lightweight and widely-used library for basic PDF operations in Python applications.

3.4 System Integration and Testing

The Chat with PDF project involves critical phases for system integration and testing to ensure a reliable and accurate document-querying experience. These phases include the development and validation of individual components such as document upload, text extraction, embedding generation, semantic search, and natural language query processing, which are then seamlessly integrated into a cohesive application. This integration process ensures efficient communication and data flow across modules, maintaining system functionality and robustness.

Unit Testing: Each component within the system undergoes rigorous unit testing to verify functionality independently. This includes testing for document upload, successful text extraction from PDFs, embedding generation, and the accuracy of responses returned to user queries. By using testing frameworks like pytest or unittest, these individual modules are validated to function as expected. Unit tests also cover input validation to handle cases like unsupported file types or empty PDF uploads, ensuring smooth interaction for users.

Integration Testing: Integration testing is conducted to validate the interoperability of modules within the Chat with PDF pipeline, ensuring each module correctly processes data passed from others in sequence. For example, the extracted text must integrate seamlessly with the embedding generation module, and embeddings should then interact correctly with the semantic search function. Integration testing guarantees that data flows properly and relevant results are returned to users' queries without interruption, enhancing user experience. Performance Testing: Performance testing assesses the system's efficiency in handling a variety of query loads and document sizes, measuring response time, memory usage, and computational requirements. Benchmarking is done to evaluate the system under various load conditions, including handling multiple PDFs and concurrent user queries. This testing phase identifies optimization opportunities, such as improving search speed in large document sets or reducing memory consumption when generating embeddings, ensuring the application remains responsive and scalable.

Error Handling and Logging: Error handling and logging mechanisms are crucial for monitoring system performance and capturing issues during runtime. Logs track actions like document uploads, query submissions, and any errors encountered during processing, enabling proactive issue detection and resolution. Logging also provides transparency in how each component performs, assisting in debugging and maintaining the system's stability in a production environment.

End-to-End Testing: End-to-end testing validates the complete functionality of the Chat with PDF system, simulating real-world scenarios. This involves uploading diverse PDFs, asking complex queries, and verifying that responses are accurate and contextually relevant. Testing includes edge cases such as ambiguous questions or documents with unusual structures to ensure robustness. Additionally, feedback from test users is solicited to confirm that the system meets usability expectations and effectively delivers information in a conversational format.

Validation and Verification: To ensure accuracy, the system's responses are validated against expected outputs, comparing results to ground-truth answers or using human annotations for complex queries. Validation confirms that responses are not only relevant but also semantically accurate, addressing the user's intent. Verification includes receiving feedback from stakeholders to confirm the system aligns with project goals, making adjustments as necessary to improve accuracy and reliability.

3.5 Deployment and Maintenances

Deploying the Chat with PDF system requires a series of carefully orchestrated steps to ensure its reliability, usability, and efficiency. The initial setup begins with the preparation of document-based data, allowing users to upload multiple PDF files to the system. The system then extracts text from these PDFs using libraries like PyPDF2, and processes the extracted content through embedding techniques to enable effective information retrieval based on user queries. This foundation allows the conversational interface to respond accurately to user inquiries by matching questions to relevant document content.

Once data extraction and processing are configured, the next phase is building a robust infrastructure for embedding generation and storage. Embeddings transform the PDF content into a searchable vector format, typically stored in a vector database such as FAISS, which enables efficient semantic searches. For large volumes of PDFs, the vector store must be optimized for performance to ensure fast, contextually accurate responses, even when handling complex or lengthy queries.

The core of the system is the language model selected for question-answering and natural language processing tasks. This model interprets user queries, retrieves relevant document content, and formulates responses. Integrating this model with the Streamlit-based interface provides users with an intuitive experience, enabling them to interact naturally with the PDFs. This conversational interface is hosted on a cloud platform or server, where deployment involves establishing a robust backend that can handle multiple concurrent requests, ensuring scalability and availability for a seamless user experience.

Deployment involves packaging the application, which includes configuring the Streamlit interface, the LangChain integration, and the model API for real-time response generation. The system can be deployed on cloud platforms like AWS, Azure, or Google Cloud, depending on project needs and scalability requirements. Establishing an API for the model allows the application to interact with the backend seamlessly, making the system capable of handling a high volume of document queries from multiple users. This deployment environment ensures that users can upload PDFs, ask questions, and receive instant, contextually relevant answers through a smooth interface.

Maintenance is a continuous task post-deployment, focusing on monitoring system performance, ensuring that text extraction, embedding generation, and document retrieval are functioning accurately and efficiently. Regular updates are essential to keep the language model, embeddings, and vector database optimized for new types of queries or additional document formats. This includes refreshing the model and retraining as necessary to maintain contextual accuracy.

The feedback loop is essential in this project, where user feedback helps identify cases where responses may need refinement. This feedback helps improve the system over time by adjusting the language model or adding new context sources. Automated monitoring and logging help track system health, document processing efficiency, and response accuracy, providing a foundation for continuous improvement. Scalability is also key as the system may need to handle increasing volumes of PDFs or concurrent users. Ensuring a scalable infrastructure allows for the addition of new features and document sources, while load balancing and resource optimization prevent performance degradation. Automating model updates and database indexing helps in maintaining high responsiveness, accuracy, and user satisfaction over time, making the Chat with PDF system a reliable and valuable tool for interactive document querying in professional and academic environments.
Ideate

IV. PROPOSED SYSTEM

4.1 Proposed System

4.1.1 Module 1: Data Collection and Preprocessing

The Chat with PDF project begins with gathering a diverse set of PDF documents, which users can upload to the system. These documents may come from various domains such as academic papers, legal files, business reports, or any other sources containing essential information. The preprocessing phase involves converting each PDF file into a uniform format for analysis, which includes extracting the text using libraries like PyPDF2. During preprocessing, irrelevant elements like headers, footers, and metadata are removed to maintain content quality and readability. This stage ensures the data is structured and standardized, laying a foundation for accurate and efficient querying within the application.

4.1.2 Module 2: Text Embedding Generation

Once the text is extracted and cleaned, the next module focuses on converting this data into embeddings, which are vector representations that capture semantic meaning. This is achieved using transformer-based models such as Sentence Transformers, which transform the text into a numerical format suitable for computational processing. These embeddings allow the system to understand and relate concepts across multiple documents, enabling it to identify contextually relevant responses to user queries. Creating high-quality embeddings is essential for accurate semantic search and ensures the AI model retrieves the most relevant information from the PDF collection.

4.1.3 Module 3: Integration with LangChain

LangChain is employed to structure and chain different AI tasks for conversational interaction, making the system capable of handling complex, multi-step queries. LangChain enables the integration of large language models (LLMs) with external data sources, such as our embedding database, and manages interactions for a smooth user experience. This module ensures that queries are effectively broken down, processed, and responded to accurately, regardless of complexity. By enabling multi-step task chaining, LangChain provides a robust framework for dynamic, contextually aware conversations with PDF documents.

4.1.4 Module 4: Building the Conversational Interface

Using Streamlit, an interactive web application is created to serve as the user interface for uploading PDFs, submitting queries, and receiving responses. This interface allows users to interact with the PDF documents naturally, typing questions in conversational language. Streamlit simplifies the process of creating a responsive, visually appealing app, providing users with instant feedback from the AI model. This module focuses on optimizing the user experience, making the document querying process accessible, intuitive, and efficient for various use cases.

4.1.5.Module5: Embedding Storage and Semantic Retrieval

To perform efficient semantic searches, the vector embeddings generated in Module 2 are stored using FAISS, a high-performance similarity search library. FAISS enables rapid retrieval of document sections most relevant to a user's query, even within large datasets. This module includes indexing and organizing embeddings to optimize search speed and accuracy. The combination of FAISS and embedding storage allows users to query vast PDF collections, ensuring the system provides the most contextually appropriate results.

4.1.6 Module 6: Model Evaluation and Validation

To ensure the system's performance, an evaluation module is set up to test accuracy, relevance, and speed of responses. Validation involves querying the model across diverse test cases and assessing its ability to retrieve correct document passages. Metrics such as retrieval accuracy, response time, and relevance to user queries are measured, providing insights into system performance. Continuous evaluation ensures that the model consistently provides valuable information, making adjustments as needed for improved accuracy.

4.1.7 Module 7: Real-Time PDF Interaction

This module enables real-time interaction with PDF documents, allowing users to continuously upload new files, ask follow-up questions, and retain conversational context. Real-time processing leverages LangChain's capability to handle dynamic interactions, so users can ask successive questions related to previous responses. The system can remember conversation history within sessions, making it a useful tool for in-depth document analysis and exploration across multiple PDFs.

4.1.8 Module 8: Reporting and Insights Generation

This module focuses on generating actionable insights and summaries based on user queries and document content. For example, after reviewing a set of documents, the system can compile summaries or provide key insights based on frequently asked questions. These reports offer users valuable takeaways, assisting them in quickly identifying essential information and making informed decisions. The reporting feature is especially beneficial for users handling extensive, complex document collections, helping them to streamline information retrieval and manage knowledge effectively.

4.2 Advantages

The Chat with PDF system leverages advanced information retrieval techniques to empower users in making well-informed decisions, improving efficiency in accessing relevant knowledge, and enhancing productivity across various fields. By transforming static PDF documents into interactive, queryable resources, this tool enables users to locate precise information quickly, thereby optimizing time and resource allocation. For organizations, this translates into streamlined workflows, where employees can instantly extract critical insights from vast document collections, aiding in strategic planning, product development, compliance checks, and decision-making processes. Educational institutions, legal firms, and research organizations also benefit as users gain rapid access to pertinent data, fostering more informed discussions and thorough analyses.

In the modern landscape, where digital documentation plays a central role in both professional and personal environments, Chat with PDF significantly enhances accessibility and usability. This tool not only allows users to query documents conversationally but also enables deeper exploration of complex topics across multiple files, making it invaluable for researchers, students, and professionals who rely on extensive literature reviews or document analysis. Furthermore, the platform's integration with metadata, such as document creation dates, author information, and subject tags, adds layers of context to retrieved information. This enriched retrieval approach not only improves precision but also provides a holistic view of the document's content, supporting a deeper understanding of the material at hand.

The ability to interact dynamically with PDF content fosters a culture of efficient knowledge management, where users can quickly retrieve and cross-reference information from various sources. This flexibility enhances productivity and informed decision-making across sectors and aligns with broader objectives of promoting a data-driven, knowledge-centric society.

V. RESULT AND SCREESNSHOTS

Output:

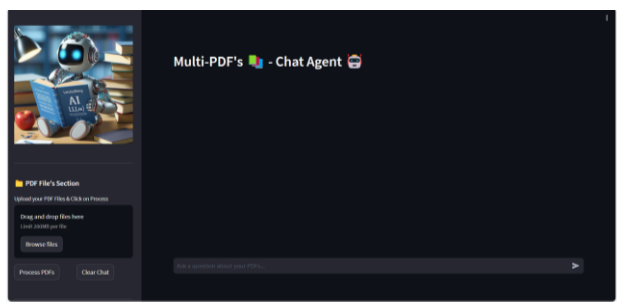

Fig 1: Output Page 1

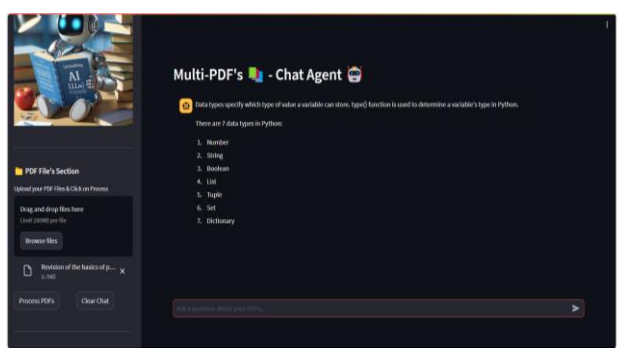

Fig 2: Output Page 2

VI. CONCLUSION

In conclusion, the use of BERT to analyse public sentiment A COVID-specific tweet has provided valuable insight public sentiment is complex and constantly changing during the prevalence pandemic. Systematically, including data collection, preprocessing, optimization of the BERT model, and real-time sensitivity analysis. We got a subtle understanding of emotion, and we got started with positivity and optimism despair and anxiety. This emotional analysis is not only It helps manage public sentiment but also allows them to determine what is important Issues and concerns expressed by the public. The ability to measure the impact of A variety of events and sensory interventions has helped here Refine public health strategies and communication efforts. Moreover, we. The approach extends beyond mere research, as we often predict Modelling, predicting potential changes in perception based on upcoming events. This prompt positioning empowers decisionmakers to respond quickly and effectively It ensures problem-solving with greater flexibility for emerging societal problems draft. Although BERT has become and continues to be a powerful tool for emotional analysis It is not without challenges. Changing models of language, addressed Biases in data, and support for ethical considerations in handling emotions remain an ongoing concern. But our determination to. Continuous improvement and the ethical application of emotional research He remains unmoved. Provide actionable reports and insights on a regular basis. Our goal for stakeholders is to help them make informed and effective decisions through public engagement throughout the COVID-19 pandemic.

VII. FUTURE SCOPE

Sentiment analysis, also known as mind mining, is a method of determining the emotions or feelings behind information, such as social media posts, reviews, customer feedback or artificial intelligence (AI) and in natural language processing With rapid advances in (NLP) technology, the future of cognitive analytics is poised for major changes, making it more accurate, efficient and insightful than ever before Traditionally, sentiment analysis has relied on specialized methods such as keyword matching, where specific words or phrases are associated with positive or negative emotions but this approach has limitations, as it is often not capturing people-based understanding of language nuances, such as sarcasm, humour, or context It also uses advanced machine learning algorithms and NLP techniques to accurately identify underlying emotions. A key advancement in AI-enabled sentiment analysis has been the use of deep learning models, such as recurrent neural networks (RNNs) and transformers These models can process large amounts of data and they recognize complex structures and relationships between words and sentences Consequently, they are better able to capture linguistic context and nuance, making sense analysis more accurate and reliable. Another important advancement in AI-powered sentiment analysis is the ability to analyse text in multiple languages. and important due to the increasing globalization of industry and the internet. Another important advancement in AI-powered sentiment analysis is the ability to analyse text in multiple languages. Globalization and the rise of the Internet have made the need for multilingual sensitivity analysis more important than ever. AI-powered sentiment analytics tools are now able to automatically search and analyse content in different languages, giving companies insights into consumer moods and sentiments across markets and regions Furthermore, AI-powered sentiment analysis tools perform well in many contexts. 33 AI-powered sentiment analysis tools can rapidly process and analyse vast amounts of data, providing real-time insights and enabling businesses to make data-driven decisions faster. With these advancements followed by AI-powered sentiment analysis Even the ability to identify and analyse emotions beyond their original categories of positive, negative and neutral. Using advanced NLP techniques and machine learning algorithms, AI-powered sentiment analysis tools can now identify and categorize a wide range of emotions such as happiness, anger, fear and shock thereby enabling businesses and researchers to navigate and understand the emotional state of their target audience and be able to make informed decisions.

REFERENCES

[1] G. G. Jayasurya, S. Kumar, B. K. Singh, and V. Kumar, "Analysis of Public Sentiment on COVID-19 Vaccination Using Twitter," in IEEE Transactions on Computational Social Systems, vol. 9, no. 4, pp. 1101-1111, Aug. 2022, doi: 10.1109/TCSS.2021.3122439.

[2] N. Rifat, M. Ahsan, R. Gomes and M. Chowdhury, "COVID-19 Sentiment Analysisapplying BERT," 2022 IEEE International Conference on Electro-Information Technology(eIT), Mankato, MN, USA, 2022, pp. 417-422, doi: 10.1109/eIT53891.2022.9813777.

[3] D. SakethNath, H. Kaur and A. Singh, "Emotion Detection using Natural Language Processing," 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 2022, pp.

[4] A. Joshy and S. Sundar, "Analyzing the Performance of Sentiment Analysis using BERT, DistilBERT, and RoBERTa," 2022 IEEE International Power and Renewable Energy Conference (IPRECON), Kollam, India, 2022, pp.

[5] R. Olusegun, T. Oladunni, H. Audu, Y. Houkpati and S. Bengesi, "Text Mining and Emotion Classification on Monkeypox Twitter Dataset: A Deep Learning-NaturalLanguage Processing (NLP) Approach," in IEEE Access, vol. 11, pp.
[6] Rustum F. Khalid M, Aslam W, Rupapara V, Mehmood A, Choi GS (2021) A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. PLoS ONE 16(2): e0245909. doi:10.1371/journal.pone.0245909
[7] İ. Aygün, B. Kaya and M. Kaya, "Aspect-Based Twitter Sentiment Analysis on Vaccination and Vaccine Types in COVID-19 Pandemic With Deep Learning," in IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 5, pp. 2360-2369, May 2022, doi:10.1109/JBHI.2021.3133103.
[8] T. Wang, K. Lu, K. P. Chow and Q. Zhu, "COVID-19 Sensing: Negative Sentiment Analysis on Social Media in China via BERT Model," in IEEE Access, vol. 8, pp. 138162138169, 2020, doi: 10.1109/ACCESS.2020.3012595.